

第四次作业

第四次作业

1. 对游戏APK的分析
2. 基于il2CppDumper分析libil2cpp.so文件
3. 实战外挂实现
 - 3.1 直接修改so文件
 - 3.2 Frida脚本编写, Hook游戏实现外挂

- 分析环境

OS	Arch
Android 6.0	i686 (4.0.9-android-x86+)
Android 6.0 (Linux kernel v3.10.0+)	armv7

- 使用工具

Tools	Version
Detect It Easy	v3.01
MuMu Player6.0	v2.7.34.0(x86)
MT Manager	v2.14.0
DnSpy	v6.1.8
IL2CppDumper	https://github.com/Perfare/IL2CppDumper 最近更新于23-3-5
frida	v16.1.5
adb	v10.0.19045

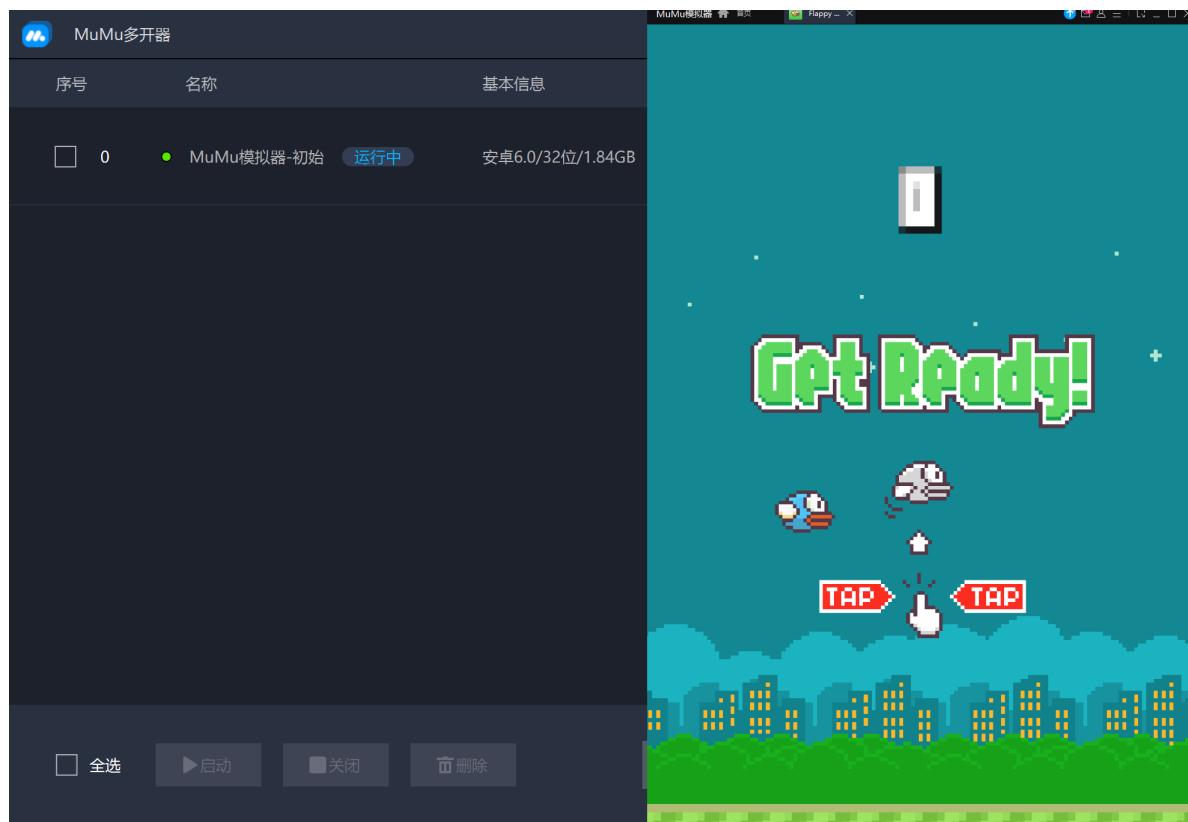
1. 对游戏APK的分析

将安装包push入MuMu12.0, 发现安卓版本太高, 在apkXML文件中有明确限制。

```
D:\Learning\2023project\TLearning\题目3和题目4\平仔骏-U202112052-第四次作业>adb install FlappyBird.apk
Performing Streamed Install
adb: failed to install FlappyBird.apk: Failure [INSTALL_PARSE_FAILED_MANIFEST_MALFORMED: Failed parse during installPackageLI: /data/app/vmdl1195189510.tmp/base.apk (at Binary XML file line #41): com.unity3d.player.UnityPlayerActivity: Targeting S+ (version 31 and above) requires that an explicit value for android:exported be defined when intent filters are present]
```

故换为MuMu6.0, 成功安装游戏, 正常运行。

```
D:\Learning\2023project\TLearning\题目3和题目4\平仔骏-U202112052-第四次作业>adb install FlappyBird.apk
Performing Push Install
FlappyBird.apk: 1 file pushed, 0 skipped. 26.4 MB/s (9370532 bytes in 0.339s)
          pkg: /data/local/tmp/FlappyBird.apk
Success
```



用MT管理器提取安装包，包名 com.lordzed.flappyzed 未加固，继续查看其XML文件和lib动态库

 **Flappy Zed**
0.1

包名 com.lordzed.flappyzed
版本号 1
安装包大小 8.94M
签名状态 V1 + V2
加固状态 未加固
已安装 0.1 (1)
数据目录 1 /data/user/0/com.lordzed.flappyzed
数据目录 2 /storage/emulated/0/Android/data/com.lordzed.flappyzed
APK 路径 /data/app/com.lordzed.flappyzed-1/base.apk
UID 10035

功能 查看 安装

XML文件没什么有用信息，只能找到主界面 com.unity3d.player.UnityPlayerActivity 这应该是Unity游戏默认的界面名字。

```

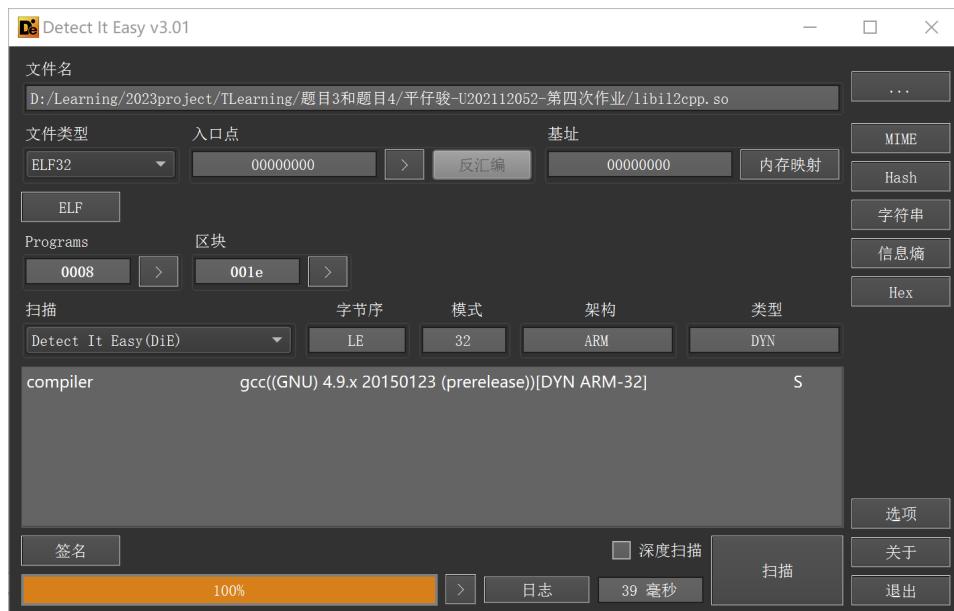
<application
    android:theme="@7F040001"
    android:label="@7F030000"
    android:icon="@7F020000"
    android:banner="@7F010000"
    android:isGame="true">
    <activity
        android:label="@7F030000"
        android:name="com.unity3d.player.UnityPlayerActivity"
        android:launchMode="2"
        android:screenOrientation="1"
        android:configChanges="0x40003FFF"
        android:hardwareAccelerated="false">
        <intent-filter>
            <action
                android:name="android.intent.action.MAIN" />
            <category
                android:name="android.intent.category.LAUNCHER" />
            <category
                android:name="android.intent.category.LEANBACK_LAUNCHER" />
        </intent-filter>
        <meta-data
            android:name="unityplayer.UnityActivity"
            android:value="true" />
    </activity>

```

lib文件夹里颇有收获，发现 `libil2cpp.so` 文件，说明该游戏使用 `unity-il2cpp` 框架。



通过DIE查询，可知是32位的可执行文件，用IDA32打开，但是没有符号信息，无法直接进行静态分析



```

File Edit Jump Search View Debugger Lumina Options Windows Help
Library function Regular function Instruction Data Unexplored External symbol Lumina function
Functions window IBA View-A Pseudocode-A Hex View-1 Structures Enums Imports Exports
Function name
Function name
59 sub_4921AA(v7, v6, 0);
51 v8 = sub_137330(dword_7683FC);
52 sub_1E6EC(v8, 1, 0);
53 if ( !v8 )
54 {
55     sub_12AD08(0);
56     sub_2B5F88(v5, v7, v8, 0);
57     if ( v7 )
58     {
59         v9 = *(__DWORD *)v7;
60         if ( *(__WORD *)(*(__DWORD *)v7 + 182) )
61         {
62             v10 = *(__WORD *)v9 + 88;
63             v11 = 0;
64             while ( *(__DWORD *)v10 + 8 * v11 != dword_767E60 )
65             {
66                 if ( !(+v11 >=(unsigned int)*(unsigned __int16 *)(*(__DWORD *)v7 + 182)) )
67                     goto LABEL_22;
68             }
69             v12 = v9 + 8 * *(__DWORD *)v10 + 8 * v11 + 4 + 192;
70         }
71     }
72 LABEL_22:
73     v12 = sub_FBD04(v7, dword_767E60, 0);
74 }
75 (*void __fastcall **)(int, __DWORD)v12(v7, *(__DWORD *)v12 + 4);
76
77 v13 = dword_768368;
78 if ( (*(__BYTE *)dword_768368 + 191) & 2 ) != 0 && !*(__DWORD *)dword_768368 + 112 )
79 {
80     il2cpp_runtime_class_init_0();
81     v13 = dword_768368;
82 }
83 *(__BYTE *)(*(__DWORD *)v13 + 92 + 13) = 1;
0019FA3C sub_19F81C:65 (19FA3C)

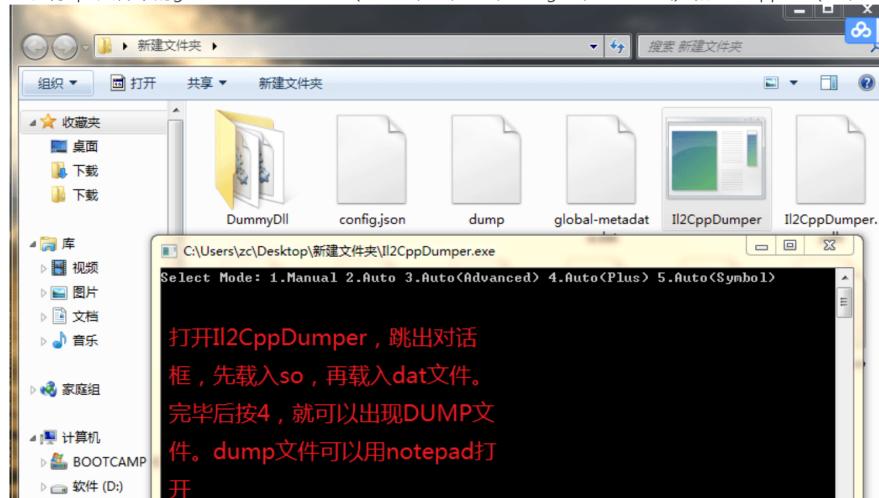
```

Line 250 of 16009

2. 基于Il2CppDumper分析libil2cpp.so文件

由于IL2CPP有静态符号信息泄露风险，我们可以使用IL2CPPDumper直接获取类名、方法名以及它们的偏移。根据Il2CppDumper的使用教程，解压apk文件并找出想dump的so文件和dat文件。

一、将apk文件中的global-metadata.dat (assets\bin\Data\Managed\Metadata\) 和libil2cpp.so (lib\armeabi-v7a\) 解压到同一个文件夹中



```

1 D:\Learning\CTF\AndroidReverse\Il2CppDumper\Il2CppDumper.exe global-
metadata.dat global-metadata.dat
2 Select Mode: 1.Manual 2.Auto 3.Auto(Advanced) 4.Auto(Plus) 5.Auto(Symbol)
3 4

```

转储下来的文件很多，我们逐个分析。先关注 DummyDll 文件夹下的 Assembly-csharp.dll，这在 windows系统下的Unity游戏中可是很重要的库文件，用 DnsSpy 打开

```

using System;
using IL2CPPDummyP1;
using UnityEngine;

// Token: 0x00000009 RID: 9
[Token(Token = "0x00000009")]
public class PlayerController : MonoBehaviour
{
    // Token: 0x0000002A RID: 42 RVA: 0x00000250 File Offset: 0x00000250
    [Token(Token = "0x00000018")]
    [Address(RVA = "0x5E22D8", Offset = "0x5E22D8", VA = "0x5E22D8")]
    public PlayerController()
    {
        // Token: 0x0000002B RID: 43 RVA: 0x00000250 File Offset: 0x00000250
        [Token(Token = "0x00000019")]
        [Address(RVA = "0x5E22E0", Offset = "0x5E22E0", VA = "0x5E22E0")]
        private void Start()
        {
            // Token: 0x0000002C RID: 44 RVA: 0x00000250 File Offset: 0x00000250
            [Token(Token = "0x0000001A")]
            [Address(RVA = "0x5E23E4", Offset = "0x5E23E4", VA = "0x5E23E4")]
            private void Update()
            {
                // Token: 0x0000002D RID: 45 RVA: 0x00000250 File Offset: 0x00000250
                [Token(Token = "0x0000001B")]
                [Address(RVA = "0x5E2788", Offset = "0x5E2788", VA = "0x5E2788")]
                private void LateUpdate()
                {
                    // Token: 0x0000002E RID: 46 RVA: 0x00000250 File Offset: 0x00000250
                    [Token(Token = "0x0000001C")]
                    [Address(RVA = "0x5E2A70", Offset = "0x5E2A70", VA = "0x5E2A70")]
                    private void OnCollisionEnter2D(Collision2D col)
                    {
                        // Token: 0x0000002F RID: 47 RVA: 0x00000250 File Offset: 0x00000250
                        [Token(Token = "0x0000001D")]
                        [Address(RVA = "0x5E30C8", Offset = "0x5E30C8", VA = "0x5E30C8")]
                        private void OnTriggerEnter2D(Collider2D col)
                        {
                            // Token: 0x00000020 RID: 48 RVA: 0x00000250 File Offset: 0x00000250
                            [Token(Token = "0x0000001E")]
                            [Address(RVA = "0x5E2788", Offset = "0x5E2788", VA = "0x5E2788")]
                            private void OnTriggerEnter2D(Collider2D col)
                            {
                                // Token: 0x00000021 RID: 49 RVA: 0x00000250 File Offset: 0x00000250
                                [Token(Token = "0x0000001F")]
                                [Address(RVA = "0x5E2A70", Offset = "0x5E2A70", VA = "0x5E2A70")]
                                private void OnTiltSmooth()
                                {
                                    // Token: 0x00000022 RID: 50 RVA: 0x00000250 File Offset: 0x00000250
                                    [Token(Token = "0x00000020")]
                                    [Address(RVA = "0x5E2A70", Offset = "0x5E2A70", VA = "0x5E2A70")]
                                    private void OnTiltSmooth()
                                    {
                                        // Token: 0x00000023 RID: 51 RVA: 0x00000250 File Offset: 0x00000250
                                        [Token(Token = "0x00000021")]
                                        [Address(RVA = "0x5E2A70", Offset = "0x5E2A70", VA = "0x5E2A70")]
                                        private void OnTiltSmooth()
                                        {
                                            // Token: 0x00000024 RID: 52 RVA: 0x00000250 File Offset: 0x00000250
                                            [Token(Token = "0x00000022")]
                                            [Address(RVA = "0x5E2A70", Offset = "0x5E2A70", VA = "0x5E2A70")]
                                            private void OnTiltSmooth()
                                            {
                                                // Token: 0x00000025 RID: 53 RVA: 0x00000250 File Offset: 0x00000250
                                                [Token(Token = "0x00000023")]
                                                [Address(RVA = "0x5E2A70", Offset = "0x5E2A70", VA = "0x5E2A70")]
                                                private void OnTiltSmooth()
                                                {
                                                    // Token: 0x00000026 RID: 54 RVA: 0x00000250 File Offset: 0x00000250
                                                    [Token(Token = "0x00000024")]
                                                    [Address(RVA = "0x5E2A70", Offset = "0x5E2A70", VA = "0x5E2A70")]
                                                    private void OnTiltSmooth()
                                                    {
                                                        // Token: 0x00000027 RID: 55 RVA: 0x00000250 File Offset: 0x00000250
                                                        [Token(Token = "0x00000025")]
                                                        [Address(RVA = "0x5E2A70", Offset = "0x5E2A70", VA = "0x5E2A70")]
                                                        private void OnTiltSmooth()
                                                        {
                                                            // Token: 0x00000028 RID: 56 RVA: 0x00000250 File Offset: 0x00000250
                                                            [Token(Token = "0x00000026")]
                                                            [Address(RVA = "0x5E2A70", Offset = "0x5E2A70", VA = "0x5E2A70")]
                                                            private void OnTiltSmooth()
                                                            {
                                                                // Token: 0x00000029 RID: 57 RVA: 0x00000250 File Offset: 0x00000250
                                                                [Token(Token = "0x00000027")]
                                                                [Address(RVA = "0x5E2A70", Offset = "0x5E2A70", VA = "0x5E2A70")]
                                                                private void OnTiltSmooth()
                                                                {
                                                                    // Token: 0x0000002A RID: 58 RVA: 0x00000250 File Offset: 0x00000250
                                                                    [Token(Token = "0x00000028")]
                                                                    [Address(RVA = "0x5E2A70", Offset = "0x5E2A70", VA = "0x5E2A70")]
                                                                    private void OnTiltSmooth()
                                                                    {
                                                                        // Token: 0x0000002B RID: 59 RVA: 0x00000250 File Offset: 0x00000250
                                                                        [Token(Token = "0x00000029")]
                                                                        [Address(RVA = "0x5E2A70", Offset = "0x5E2A70", VA = "0x5E2A70")]
                                                                        private void OnTiltSmooth()
                                                                        {
                                                                            // Token: 0x0000002C RID: 60 RVA: 0x00000250 File Offset: 0x00000250
                                                                            [Token(Token = "0x0000002A")]
                                                                            [Address(RVA = "0x5E2A70", Offset = "0x5E2A70", VA = "0x5E2A70")]
                                                                            private void OnTiltSmooth()
                                                                            {
                                                                                // Token: 0x0000002D RID: 61 RVA: 0x00000250 File Offset: 0x00000250
                                                                                [Token(Token = "0x0000002B")]
                                                                                [Address(RVA = "0x5E2A70", Offset = "0x5E2A70", VA = "0x5E2A70")]
                                                                                private void OnTiltSmooth()
                                                                                {
                                                                                    // Token: 0x0000002E RID: 62 RVA: 0x00000250 File Offset: 0x00000250
                                                                                    [Token(Token = "0x0000002C")]
                                                                                    [Address(RVA = "0x5E2A70", Offset = "0x5E2A70", VA = "0x5E2A70")]
                                                                                    private void OnTiltSmooth()
                                                                                    {
                                                                                        // Token: 0x0000002F RID: 63 RVA: 0x00000250 File Offset: 0x00000250
                                                                                        [Token(Token = "0x0000002D")]
                                                                                        [Address(RVA = "0x5E2A70", Offset = "0x5E2A70", VA = "0x5E2A70")]
                                                                                        private void OnTiltSmooth()
                                                                                        {
                                                                                            // Token: 0x00000030 RID: 64 RVA: 0x00000250 File Offset: 0x00000250
                                                                                            [Token(Token = "0x0000002E")]
                                                                                            [Address(RVA = "0x5E2A70", Offset = "0x5E2A70", VA = "0x5E2A70")]
                                                                                            private void OnTiltSmooth()
                                                                                            {
                                                                                                // Token: 0x00000031 RID: 65 RVA: 0x00000250 File Offset: 0x00000250
                                                                                                [Token(Token = "0x0000002F")]
                                                                                                [Address(RVA = "0x5E2A70", Offset = "0x5E2A70", VA = "0x5E2A70")]
                                                                                                private void OnTiltSmooth()
                                                                                                {
                                                                                                 
                                                                                                 
................................................................

```

打开发现有类名也有方法名，但是里面的实现都是空的，其中 `PlayerController` 很明显是控制玩家对象的类，`OnCollisionEnter2D` 和 `OnTriggerEnter2D` 很明显是翻越管道的加分逻辑和碰撞反馈的逻辑函数，后面应该会重点研究。

通过右侧去掉实现内容的代码信息中，也能观察到每个成员的偏移地址，后面可能会用到。

```

using System;
using IL2CPPDummyP1;
using UnityEngine;

// Token: 0x00000007 RID: 7 RVA: 0x00000204 File Offset: 0x00000204
[Token(Token = "0x00000007")]
public void UpdateScore()
{
}

// Token: 0x00000008 RID: 8 RVA: 0x00000204 File Offset: 0x00000204
[Token(Token = "0x00000008")]
[Address(RVA = "0x5E000008", Offset = "0x5E000008", VA = "0x5E000008")]
public bool GameState()
{
    return default(bool);
}

// Token: 0x00000009 RID: 9 RVA: 0x00000204 File Offset: 0x00000204
[Token(Token = "0x00000009")]
[Address(RVA = "0x5E000009", Offset = "0x5E000009", VA = "0x5E000009")]
public void EndGame()
{
    return null;
}

// Token: 0x0000000A RID: 10 RVA: 0x00000204 File Offset: 0x00000204
[Token(Token = "0x0000000A")]
[Address(RVA = "0x5E00000A", Offset = "0x5E00000A", VA = "0x5E00000A")]
public void Replay()
{
    return null;
}

// Token: 0x0000000B RID: 11 RVA: 0x00000204 File Offset: 0x00000204
[Token(Token = "0x0000000B")]
[Address(RVA = "0x5E00000B", Offset = "0x5E00000B", VA = "0x5E00000B")]
private IEnumerator StartGame()
{
    return null;
}

// Token: 0x0000000C RID: 12 RVA: 0x00000204 File Offset: 0x00000204
[Token(Token = "0x0000000C")]
[Address(RVA = "0x5E00000C", Offset = "0x5E00000C", VA = "0x5E00000C")]
private void Leaderboard()
{
}

```

`GameManager` 类同样需要重点关注，因为里面有许多控制分数和游戏状态的逻辑，如 `updateScore`、`UpdateScore` 和 `GameState` 等方法。

接下来使用 `IDA32`，结合 `IL2CPPDumper` 仓库中现有的 python 脚本，恢复 `i12cpp.h` 中的结构体信息和 `script.json` 中的符号信息，从而静态分析 `libi12cpp.so` 文件。

DummyDll
文件夹, 包含所有还原的DLL文件
使用dnSpy, ILSPy或者其他Net反编译工具即可查看具体信息
可用于提取Unity的MonoBehaviour和MonoScript, 适用于UtinyRipper或者UABE等

ida.py
用于IDA
ida_with_struct.py
用于IDA, 读取il2cpp.h文件并在IDA中应用结构信息

il2cpp.h
包含结构体的头文件

ghidra.py
用于Ghidra

Il2CppBinaryNinja
用于BinaryNinja

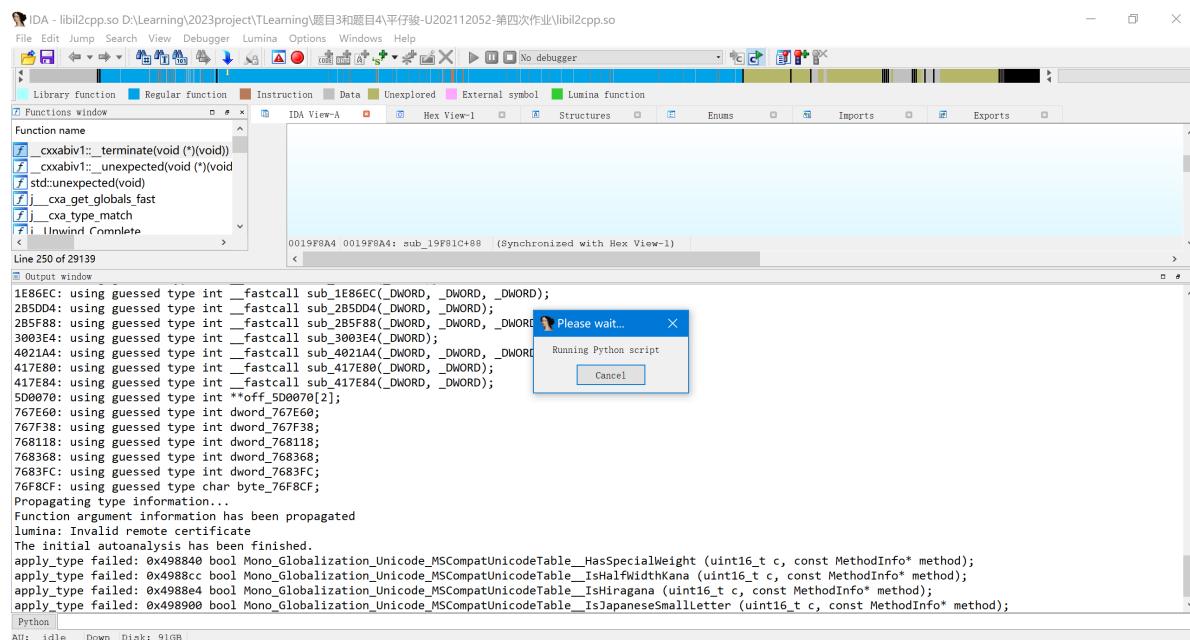
ghidra_wasm.py
用于Ghidra, 和ghidra-wasm-plugin一起工作

script.json
用于IDA和Ghidra脚本

Data (D) > Learning > CTF > AndroidReverse > Il2CppDumper > Il2CppDumper

名称	修改日期	类型	大小
Attributes	2023-11-07 22:42	文件夹	
ExecutableFormats	2023-11-07 22:42	文件夹	
Extensions	2023-11-07 22:42	文件夹	
Il2Cpp	2023-11-07 22:42	文件夹	
Il2CppBinaryNinja	2023-11-07 22:42	文件夹	
IO	2023-11-07 22:42	文件夹	
Libraries	2023-11-07 22:42	文件夹	
Outputs	2023-11-07 22:42	文件夹	
Utils	2023-11-07 22:42	文件夹	
Config.cs	2023-11-07 22:42	C# 源文件	1 KB
config.json	2023-11-07 22:42	JSON 源文件	1 KB
ghidra.py	2023-11-07 22:42	Python 源文件	3 KB
ghidra_wasm.py	2023-11-07 22:42	Python 源文件	4 KB
ghidra_with_struct.py	2023-11-07 22:42	Python 源文件	6 KB
hopper-py3.py	2023-11-07 22:42	Python 源文件	1 KB
ida.py	2023-11-07 22:42	Python 源文件	3 KB
ida_py3.py	2023-11-07 22:42	Python 源文件	3 KB
ida_with_struct.py	2023-11-07 22:42	Python 源文件	3 KB
ida_with_struct_py3.py	2023-11-07 22:42	Python 源文件	3 KB
il2cpp_header_to_binja.py	2023-11-07 22:42	Python 源文件	2 KB
il2cpp_header_to_ghidra.py	2023-11-07 22:42	Python 源文件	2 KB

最后选择 ida_with_struct_py3.py 脚本:



在进行符号转换的文件中搜索“PlayerController”字符串，并在结果中检索与我们需要修改游戏逻辑的几个成员属性和函数。

Address	Function	Instruction
il2cpp:002C9F18	sub_2C9F18	sub_2C9F18 ; CODE XREF: PlayerController\$\$Update...
il2cpp:002CA05C	UnityEngine.Quaternion\$\$L...	UnityEngine.Quaternion\$\$Lerp ; CODE XREF: PlayerController\$\$...
il2cpp:002CA5E4	UnityEngine.Quaternion\$\$E...	UnityEngine.Quaternion\$\$Euler ; CODE XREF: PlayerController\$\$...
il2cpp:002D7300	UnityEngine.Transform\$\$Se...	UnityEngine.Transform\$\$Set_rotation ; CODE XREF: PlayerController...
il2cpp:002D7D50	UnityEngine.Transform\$\$G...	UnityEngine.Transform\$\$GetEnumerator ; CODE XREF: PlayerController...
il2cpp:00317FE0	UnityEngine.Component\$\$...	UnityEngine.Component\$\$CompareTag ; CODE XREF: PlayerController...
il2cpp:005E0860	GameManager\$\$GetReady	GameManager\$\$GetReady ; CODE XREF: PlayerController\$\$...
il2cpp:005E098C	GameManager\$\$UpdateSc...	GameManager\$\$UpdateScore ; CODE XREF: PlayerController\$\$...
il2cpp:005E0CD8	GameManager\$\$EndGame	GameManager\$\$EndGame ; CODE XREF: PlayerController\$\$...
il2cpp:005E22D8	PlayerController\$\$ctor	; void __fastcall PlayerController____ctor(PlayerController_o *this, const ...)
il2cpp:005E22E0	PlayerController\$\$Start	; void __fastcall PlayerController____Start(PlayerController_o *this, const ...)
il2cpp:005E2324	PlayerController\$\$Start	loc_5E2324 ; CODE XREF: PlayerController\$\$Start+241j
il2cpp:005E236C	PlayerController\$\$Start	loc_5E236C ; CODE XREF: PlayerController\$\$Start+781j
il2cpp:005E23D0	PlayerController\$\$Start	; DATA XREF: PlayerController\$\$Start+141r
il2cpp:005E23D4	PlayerController\$\$Start	; DATA XREF: PlayerController\$\$Start+281r
il2cpp:005E23D8	PlayerController\$\$Start	; DATA XREF: PlayerController\$\$Start+381r
il2cpp:005E23DC	PlayerController\$\$Start	; DATA XREF: PlayerController\$\$Start loc_5E23...
il2cpp:005E23E0	PlayerController\$\$Start	; DATA XREF: PlayerController\$\$Start+601r
il2cpp:005E23E4	PlayerController\$\$Update	; void __fastcall PlayerController____Update(PlayerController_o *this, const ...)
il2cpp:005E242C	PlayerController\$\$Update	loc_5E242C ; CODE XREF: PlayerController\$\$Update+...
il2cpp:005E2468	PlayerController\$\$Update	loc_5E2468 ; CODE XREF: PlayerController\$\$Update+...

下面是更新游戏分数的逻辑。有些 `stringLiteral_xxxx` 的 Tag 我没有用脚本进行转化，这里给出部分对应表：

在对应表中并没有 Pipe，只有 Obstacle，表示障碍物，这里 Obstacle 应该代指 Pipe

1	3004	Start
2	3005	Point
3	3006	Score
4	3007	GameOver
5	3008	Hit
6	3009	StartGame
7	3010	Swoosh
8	3011	End
9	3012	Game
10	3013	Flap
11	3014	Obstacle
12	3015	Ground

```

7 System_String_o *v7; // r0
8 const MethodInfo *v8; // r2
9 int32_t v9[3]; // [sp+4h] [bp-Ch] BYREF
10
11 if ( !byte_771C01 )
12 {
13     sub_101850(&stru_C00.st_info);
14     byte_771C01 = 1;
15 }
16 v3 = this->fields.gameScoreText;
17 v4 = this->fields.gameScore + 1;
18 this->fields.gameScore = v4;
19 v9[0] = v4;
20 v5 = (Il2CppObject *)il2cpp_value_box_0(int_TypeInfo, v9);
21 v6 = string_TypeInfo;
22 if ( (string_TypeInfo->_2.bitflags2 & 2) != 0 && !string_TypeInfo->_2cctor_started )
23 {
24     il2cpp_runtime_class_init_0();
25     v6 = string_TypeInfo;
26 }
27 v7 = System_String_Concat(v5, (Il2CppObject *)v6->static_fields->Empty, 0);
28 if ( !v3 )
29     sub_12AD08(0);
005E0A00 GameManager$$UpdateScore:17 (5E0A00)
    
```

下面是 `OnTriggerEnter2D` 的反编译代码，可以看出是游戏结束时的触发逻辑，在游戏结束时更新分数，最后还会 `KillPlayer`！

IDA View-A Pseudocode-A Occurrences of: PlayerController Hex View-1 Structures Enums Imports

```

34
35     if ( !byte_771C14 )
36     {
37         sub_101850(4885);
38         byte_771C14 = 1;
39     }
40     if ( !col )
41         sub_12AD08(0);
42     v5 = (UnityEngine_Component_o *)UnityEngine_Component__get_transform((UnityEngine_Component_o *)col, 0);
43     if ( !v5 )
44         sub_12AD08(0);
45     if ( UnityEngine_Component__CompareTag(v5, (System_String_o *)StringLiteral_3006, 0) )
46     {
47         v6 = (UnityEngine_Object_o *)UnityEngine_Component__get_gameObject((UnityEngine_Component_o *)col, 0);
48         if ( (UnityEngine_Object_TypeInfo->_2.bitflags2 & 2) != 0 && !UnityEngine_Object_TypeInfo->_2cctor_started )
49             il2cpp_runtime_class_init_0();
50         UnityEngine_Object__Destroy_2910632(v6, 0);
51         if ( !GameManager_TypeInfo->static_fields->Instance )
52             sub_12AD08(0);
53         GameManager__UpdateScore(GameManager_TypeInfo->static_fields->Instance, v7);
54     }
55     else
56     {
57         v8 = (UnityEngine_Component_o *)UnityEngine_Component__get_transform((UnityEngine_Component_o *)col, 0);
58         if ( !v8 )
59             sub_12AD08(0);
60         if ( UnityEngine_Component__CompareTag(v8, (System_String_o *)StringLiteral_3014, 0) )
61         {
62             v9 = UnityEngine_Component__get_transform((UnityEngine_Component_o *)col, 0);

```

005E2B8C PlayerController\$\$OnTriggerEnter2D:53 (5E2B8C)

IDA View-A Pseudocode-A Occurrences of: PlayerController Hex View-1 Structures Enums

```

135     UnityEngine_Behaviour__set_enabled((UnityEngine_Behaviour_o *)v27, 0, 0);
136 }
137 v28 = (_DWORD *)sub_1375DC(v12, System_IDisposable_TypeInfo);
138 v30 = v28;
139 if ( v28 )
140 {
141     v31 = *v28;
142     if ( (*(_WORD *)(*v30 + 182) )
143     {
144         v32 = (*_DWORD *)(v31 + 88);
145         v33 = 0;
146         while ( (*(System_IDisposable_c **)(v32 + 8 * v33) != System_IDisposable_TypeInfo )
147         {
148             if ( ++v33 >= (unsigned int)*(unsigned __int16 *)(*v30 + 182) )
149                 goto LABEL_54;
150         }
151         v34 = v31 + 8 * *( _DWORD *)(v32 + 8 * v33 + 4) + 192;
152     }
153     else
154     {
155     LABEL_54:
156         v34 = sub_FBD04(v30, System_IDisposable_TypeInfo, 0);
157     }
158     (*(void (__fastcall **)(_DWORD *, _DWORD))v34)(v30, *( _DWORD *)(v34 + 4));
159 }
160 PlayerController_KillPlayer(this, v29);
161 }
162 }
```

下面是OnCollisionEnter2D的反编译代码，可以看出这里主要判断与地面的碰撞

```

12     if ( !byte_771C15 )
13     {
14         sub_101850(&stru_1310.st_value);
15         byte_771C15 = 1;
16     }
17     if ( !col )
18         sub_12AD08(0);
19     v5 = (UnityEngine_Component_o *)UnityEngine_Collision2D__get_transform(col, 0);
20     if ( !v5 )
21         sub_12AD08(0);
22     if ( UnityEngine_Component__CompareTag(v5, (System_String_o *)String_Ground, 0) )
23     {
24         if ( !this->fields.playerRigid )
25             sub_12AD08(0);
26         UnityEngine_Rigidbody2D__set_simulated(this->fields.playerRigid, 0, 0);
27         PlayerController_KillPlayer(this, v6);
28         v7 = UnityEngine_Component__get_transform((UnityEngine_Component_o *)this, 0);
29         v8 = this->fields.downRotation.fields.x;
30         v9 = this->fields.downRotation.fields.y;
31         v10 = this->fields.downRotation.fields.z;
32         v11 = this->fields.downRotation.fields.w;
33         if ( !v7 )
34             sub_12AD08(0);
35         v12.fields.x = v8;
36         *( _QWORD *)&v12.fields.y = __PAIR64__(LODWORD(v10), LODWORD(v9));
37         v12.fields.w = v11;
38         UnityEngine_Transform__set_rotation(v7, v12, 0);
39     }
40 }
```

005E3148 PlayerController\$\$OnCollisionEnter2D:22 (5E3148)

3. 实战外挂实现

3.1 直接修改so文件

回顾上面找到的3个关键方法函数，只需要用IDA Pro进行一些Patch就可以实现外挂。

- Patch分数更新逻辑，使得一次跳跃就能加255分！

```
i12cpp:005E09FC          loc_5E09FC           ; CODE XREF: GameManager$$UpdateScore+201j
i12cpp:005E09FC E4 00 9F E5      LDR      R0, =(off_75869C - 0x5E0A10)
i12cpp:005E09FC 50 10 95 E5      LDR      R1, [R5,#0x50]
i12cpp:005E0A00 20 40 95 E5      LDR      R4, [R5,#0x20]
i12cpp:005E0A04 00 00 9F E7      LDR      R0, [PC,R0] ; off_75869C ; int_TypeInfo
i12cpp:005E0A08 FF 10 81 E2      ADD      R1, R1, #0xFF ; 分数更新
i12cpp:005E0A10 50 10 85 E5      STR      R1, [R5,#0x50]
i12cpp:005E0A14 04 10 8D E5      STR      R1, [SP,#0x10+var_C]
i12cpp:005E0A18 04 10 8D E2      ADD      R1, SP, #0x10+var_C
i12cpp:005E0A1C 00 00 90 E5      LDR      R0, [R0] ; int_TypeInfo
i12cpp:005E0A20 1D 5A ED EB      BL       il2cpp_value_box_0
i12cpp:005E0A24 00 50 A0 E1      MOV      R5, R0
i12cpp:005E0A28 BC 00 9F E5      LDR      R0, =(off_75833C - 0x5E0A34)
i12cpp:005E0A2C 00 00 9F E7      LDR      R0, [PC,R0] ; off_75833C ; string_TypeInfo
i12cpp:005E0A30 00 00 90 E5      LDR      R0, [R0] ; string_TypeInfo
i12cpp:005E0A34 BF 10 D0 E5      LDRB     R1, [R0,#0xBF]
i12cpp:005E0A38 02 00 11 E3      TST      R1, #2
i12cpp:005E0A3C 06 00 00 0A      BEQ      loc_5E0A5C
i12cpp:005E0A40 70 10 90 E5      LDR      R1, [R0,#0x70]
i12cpp:005E0A44 00 00 51 E3      CMP      R1, #0
i12cpp:005E0A48 03 00 00 1A      BNE      loc_5E0A5C
```

- Patch `OnTrigger` 碰撞逻辑，使得即使撞到了障碍物也不会跳到 `KillPlayer` 逻辑，由于两次比较的Tag分别为 `GameOver` 和 `Obstacle`，需要将第二次分支判断Patch掉才能达到无敌的效果。

游戏结束也分为成功结束和失败结束，这个方法用一个嵌套的方式编写可能就是为了区分是通关后的GameOver还是碰撞了Obstacle后的GameOver

```
45 if ( UnityEngine_Component__CompareTag(v5, (System_String_o *)GameOver, 0) )
46 {
47     v6 = (UnityEngine_Object_o *)UnityEngine_Component__get_gameObject((UnityEngine_Component_o *)col, 0);
48     if ( (UnityEngine_Object_TypeInfo->_2.bitflags2 & 2) != 0 && !UnityEngine_Object_TypeInfo->_2cctor_started )
49         il2cpp_runtime_class_init_0();
50     UnityEngine_Object__Destroy_2910632(v6, 0);
51     if ( !GameManager_TypeInfo->static_fields->Instance )
52         sub_12AD08(0);
53     GameManager__UpdateScore(GameManager_TypeInfo->static_fields->Instance, v7);
54 }
55 else
56 {
57     v8 = (UnityEngine_Component_o *)UnityEngine_Component__get_transform((UnityEngine_Component_o *)col, 0);
58     if ( !v8 )
59         sub_12AD08(0);
60     if ( UnityEngine_Component__CompareTag(v8, (System_String_o *)Obstacle, 0) )
61     {
62         v9 = UnityEngine_Component__get_transform((UnityEngine_Component_o *)col, 0);
63         if ( !v9 )
64             sub_12AD08(0);
65         v10 = (UnityEngine_Component_o *)UnityEngine_Transform__get_parent(v9, 0);
```

下面是第一处判断的逻辑：

```
i12cpp:005E2AF4 06 00 A0 E1      MOV      R0, R6 ; this
i12cpp:005E2AF8 38 D5 F4 EB      BL       UnityEngine.Component$$CompareTag ; GameOver
i12cpp:005E2AFC 00 60 A0 E1      MOV      R6, R0
i12cpp:005E2B00 00 00 55 E3      CMP      R5, #0
i12cpp:005E2B04 01 00 00 1A      BNE      loc_5E2B10
i12cpp:005E2B08 00 00 A0 E3      MOV      R0, #0
i12cpp:005E2B0C 7D 20 ED EB      BL       sub_12AD08
i12cpp:005E2B10 ; -----
i12cpp:005E2B10          loc_5E2B10           ; CODE XREF: PlayerController$$OnTriggerEnter2D+94↑j
i12cpp:005E2B10 01 00 56 E3      CMP      R6, #1
i12cpp:005E2B14 1D 00 00 1A      BNE      Compare_Obstacle ; GameOver的判断逻辑
i12cpp:005E2B18 05 00 A0 E1      MOV      R0, R5 ; this
i12cpp:005E2B1C 00 10 A0 E3      MOV      R1, #0 ; method
```

第二处判断跳转的程序段在 `KillPlayer` 之后，这说明应该在这里无条件跳转，使玩家避免死亡，我们将其patch成 B 也就是将操作数机器码改为 EA

```

il2cpp:005E2BB0      loc_5E2BB0          ; CODE XREF: PlayerController$$OnTriggerEnter2D+134↑j
il2cpp:005E2BB0 D0 03 9F E5    LDR     R0, =(off_766DBC - 0x5E2BC0)
il2cpp:005E2BB4 00 20 A0 E3    MOV     R2, #0 ; method
il2cpp:005E2BB8 00 00 9F E7    LDR     R0, [PC,R0] ; off_766DBC ; Obstacle
il2cpp:005E2BBC 00 10 90 E5    LDR     R1, [R0] ; tag
il2cpp:005E2BC0 06 00 A0 E1    MOV     R0, R6 ; this
il2cpp:005E2BC4 05 D5 F4 EB    BL      UnityEngine.Component$$CompareTag ; Obstacle
il2cpp:005E2BC8 01 00 50 E3    CMP     R0, #1
il2cpp:005E2BC0 06 00 A0 E1    BNE    Alive ; no kill
il2cpp:005E2BBC 05 E5 00 1A    CMP     R5, #0
il2cpp:005E2BD0 00 00 55 E3    BNE    loc_5E2BE0
il2cpp:005E2BD4 01 00 00 1A    MOV     R0, #0
il2cpp:005E2BD8 00 00 A0 E3    BL     sub_12AD08
il2cpp:005E2BDC 49 20 ED EB

```

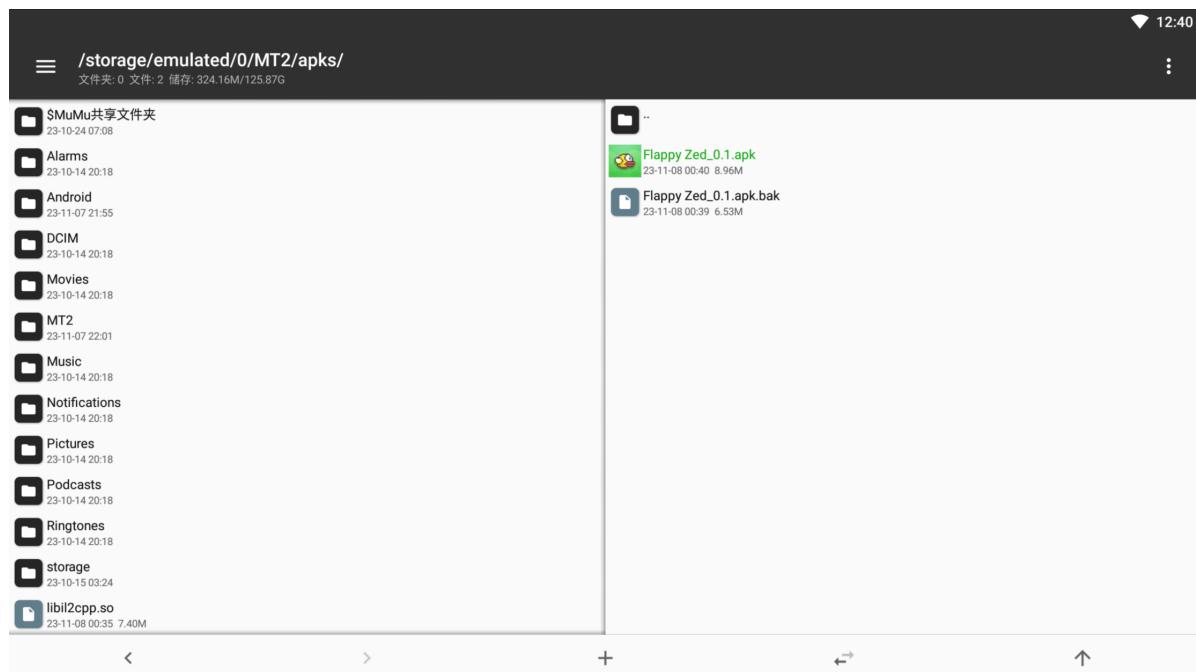
- 同上，对于触碰地面的逻辑也进行了Patch，使小鸟不会死亡。

```

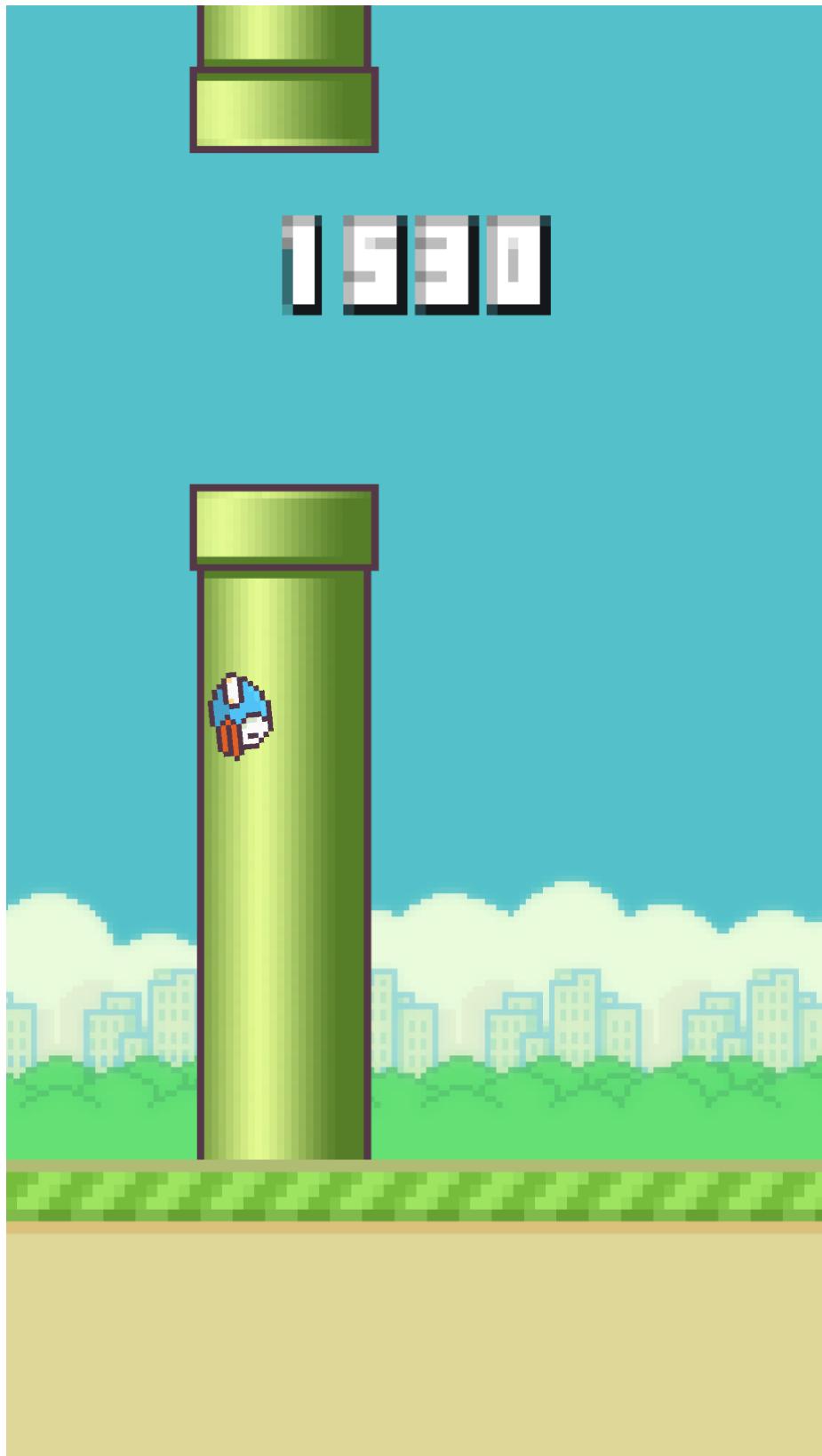
il2cpp:005E313C      loc_5E313C          ; CODE XREF: PlayerController$$OnCollisionEnter2D+68↑j
il2cpp:005E313C A0 00 9F E5    LDR     R0, =(off_766DD0 - 0x5E314C)
il2cpp:005E3140 00 20 A0 E3    MOV     R2, #0 ; method
il2cpp:005E3144 00 00 9F E7    LDR     R0, [PC,R0] ; off_766DD0 ; Ground
il2cpp:005E3148 00 10 90 E5    LDR     R1, [R0] ; tag
il2cpp:005E314C 05 00 A0 E1    MOV     R0, R5 ; this
il2cpp:005E3150 A2 D3 F4 EB    BL      UnityEngine.Component$$CompareTag ; Ground
il2cpp:005E3154 01 00 50 E3    CMP     R0, #1
il2cpp:005E3158 1C 00 00 EA    B      Alive ; 无条件跳转
il2cpp:005E315C           ; -----
il2cpp:005E315C 30 50 94 E5    LDR     R5, [R4,#0x30]
il2cpp:005E3160 00 00 55 E3    CMP     R5, #0
il2cpp:005E3164 01 00 00 1A    BNE    loc_5E3170
il2cpp:005E3168 00 00 A0 E3    MOV     R0, #0
il2cpp:005E316C E5 1E ED EB    BL     sub_12AD08
il2cpp:005E3170           ;

```

将修改后的.so文件push到模拟器中，再用MT管理器重新打包。（自动重新签名）



效果如下图，小鸟已经无视碰撞事件了，并且每次翻越能加255分。



3.2 Frida脚本编写，Hook游戏实现外挂

frida-server-16.1.5-android-arm.xz	6.62 MB	3 days ago
frida-server-16.1.5-android-arm64.xz	14.8 MB	3 days ago
frida-server-16.1.5-android-x86.xz	15 MB	3 days ago
frida-server-16.1.5-android-x86_64.xz	30.2 MB	3 days ago
frida-server-16.1.5-freebsd-arm64.xz	7.39 MB	3 days ago
frida-server-16.1.5-freebsd-x86_64.xz	7.52 MB	3 days ago

将下载好的合适版本的 frida_server push进模拟器

```
shell@x86:/ $ su
root@x86:/ # cd /data/local/tmp
root@x86:/data/local/tmp # ;ls
sh: syntax error: ';' unexpected
[1] root@x86:/data/local/tmp # ls
CrackMe2
busybox
frida-server-16.1.5-android-x86
injector
libhook.so
/frida-server-16.1.5-android-x86
sh: ./frida-server-16.1.5-android-x86: can't execute: Permission denied <
chmod 777 frida-server-16.1.5-android-x86 <
root@x86:/data/local/tmp # ./frida-server-16.1.5-android-x86

root@x86:/data/local/tmp # ./frida-server-16.1.5-android-x86 -U
Unknown option -U
Run './frida-server-16.1.5-android-x86 --help' to see a full list of available command line options.
[1] root@x86:/data/local/tmp # ./frida-server-16.1.5-android-x86
```

用 frida-ps -U 测试，可以正常运行

```
D:\Learning\2023project\TLearning\题目3和题目4\平仔骏-U202112052-第四次作业>adb shell getprop ro.product.cpu.abi
x86
D:\Learning\2023project\TLearning\题目3和题目4\平仔骏-U202112052-第四次作业>frida-ps -U
PID Name
-----
```

PID	Name
17263	Flappy_Zed
245	VBoxShell
1006	android.process.acore
887	android.process.media
1368	com.android.defcontainer
16198	com.android.externalstorage
16134	com.android.packageinstaller
1430	com.android.phone
16215	com.android.shell
788	com.android.systemui
300	debuggerd
302	drmserver
17387	frida-server-16.1.5-android-x86
345	gatekeeperd
282	healthd
1	init
304	installd
305	keystore
284	lmdk
290	local_camera_back
289	local_gps
17389	logcat
143	logd
276	logwrapper

编写frida插件脚本，修改在 3.1 中的几个地方的内存，即可实现外挂。

- [0xE3158]处的shell code

```
il2cpp:005E313C loc_5E313C ; CODE XREF: PlayerController$$OnCollisionEnter2D+68↑j
. il2cpp:005E313C A0 00 9F E5 LDR R0, =(off_766DD0 - 0xE314C)
. il2cpp:005E3140 00 20 A0 E3 MOV R2, #0 ; method
. il2cpp:005E3144 00 00 9F E7 LDR R0, [PC,R0] ; off_766DD0 ; Ground
. il2cpp:005E3148 00 10 90 E5 LDR R1, [R0] ; tag
. il2cpp:005E314C 05 00 A0 E1 MOV R0, R5 ; this
. il2cpp:005E3150 A2 D3 F4 EB BL UnityEngine.Component$$CompareTag ; Ground
. il2cpp:005E3154 01 00 50 E3 CMP R0, #1
. il2cpp:005E3158 1C 00 00 EA B Alive ; 无条件跳转
il2cpp:005E315C ; -----
. il2cpp:005E315C 30 50 94 E5 LDR R5, [R4,#0x30]
. il2cpp:005E3160 00 00 55 E3 CMP R5, #0
. il2cpp:005E3164 01 00 00 1A BNE loc_5E3170
. il2cpp:005E3168 00 00 A0 E3 MOV R0, #0
. il2cpp:005E316C E5 1E ED EB BL sub_12AD08
il2cpp:005E3170 ; -----
```

- [0xE2BCC]处的shell code

```

il2cpp:005E2BB0          loc_5E2BB0           ; CODE XREF: PlayerController$$OnTriggerEnter2D+134↑j
il2cpp:005E2BB0 D0 03 9F E5    LDR      R0, =(off_766DBC - 0x5E2BC0)
il2cpp:005E2BB4 00 20 A0 E3    MOV      R2, #0 ; method
il2cpp:005E2BB8 00 00 9F E7    LDR      R0, [PC,R0] ; off_766DBC ; Obstacle
il2cpp:005E2BBC 00 10 90 E5    LDR      R1, [R0] ; tag
il2cpp:005E2BC0 06 00 A0 E1    MOV      R0, R6 ; this
il2cpp:005E2BC4 05 D5 F4 EB    BL       UnityEngine.Component$$CompareTag ; Obstacle
il2cpp:005E2BC8 01 00 50 E3    CMP      R0, #1
il2cpp:005E2BCC E5 00 00 1A    BNE      Alive ; no kill
il2cpp:005E2BD0 00 00 55 E3    CMP      R5, #0
il2cpp:005E2BD4 01 00 00 1A    BNE      loc_5E2BE0
il2cpp:005E2BD8 00 00 A0 E3    MOV      R0, #0
il2cpp:005E2BDC 49 20 ED EB    BL       sub_12AD08

```

- [0x5E0A0C]处的shell code

```

il2cpp:005E09FC          loc_5E09FC           ; CODE XREF: GameManager$$UpdateScore+20↑j
il2cpp:005E09FC E4 00 9F E5    LDR      R0, =(off_75869C - 0x5E0A10)
il2cpp:005E0A00 50 10 95 E5    LDR      R1, [R5,#0x50]
il2cpp:005E0A04 20 40 95 E5    LDR      R4, [R5,#0x20]
il2cpp:005E0A08 00 00 9F E7    LDR      R0, [PC,R0] ; off_75869C ; int_TypeInfo
il2cpp:005E0A0C FF 10 81 E2    ADD      R1, R1, #0xFF ; 分数更新
il2cpp:005E0A10 50 10 85 E5    STR      R1, [R5,#0x50]
il2cpp:005E0A14 04 10 8D E5    STR      R1, [SP,#0x10+var_C]
il2cpp:005E0A18 04 10 8D E2    ADD      R1, SP, #0x10+var_C
il2cpp:005E0A1C 00 00 90 E5    LDR      R0, [R0] ; int_TypeInfo
il2cpp:005E0A20 1D 5A ED EB    BL       il2cpp_value_box_0
il2cpp:005E0A24 00 50 A0 E1    MOV      R5, R0
il2cpp:005E0A28 BC 00 9F E5    LDR      R0, =(off_75833C - 0x5E0A34)
il2cpp:005E0A2C 00 00 9F E7    LDR      R0, [PC,R0] ; off_75833C ; string_TypeInfo
il2cpp:005E0A30 00 00 90 E5    LDR      R0, [R0] ; string_TypeInfo
il2cpp:005E0A34 BF 10 D0 E5    LDRB     R1, [R0,#0xBF]
il2cpp:005E0A38 02 00 11 E3    TST      R1, #2
il2cpp:005E0A3C 06 00 00 0A    BEQ      loc_5E0A5C
il2cpp:005E0A40 70 10 90 E5    LDR      R1, [R0,#0x70]
il2cpp:005E0A44 00 00 51 E3    CMP      R1, #0
il2cpp:005E0A48 03 00 00 1A    BNE      loc_5E0A5C

```

```

1 // 以下脚本可以实现无视碰撞无敌
2 Java.perform(() => {
3     var libil2cpp = Process.findModuleByName("libil2cpp.so");
4     // OnTriggerEnter2D
5     var offset1 = 0x5E2BCC;
6     var addr1 = libil2cpp.base.add(offset1);
7     var arr1 = [0xE5, 0x00, 0x00, 0xEA]; // B 0x5E2BCC
8     // 修改内存保护
9     Memory.protect(addr1, 0x1000, 'rwx');
10    Memory.writeByteArray(addr1, arr1);
11    // OnCollisionEnter2D
12    var offset2 = 0x5E3158;
13    var addr2 = libil2cpp.base.add(offset2);
14    var arr2 = [0x1A, 0x00, 0x00, 0xEA]; // B 0x5E3158
15    // 修改内存保护
16    Memory.protect(addr2, 0x1000, 'rwx');
17    Memory.writeByteArray(addr2, arr2);
18 });

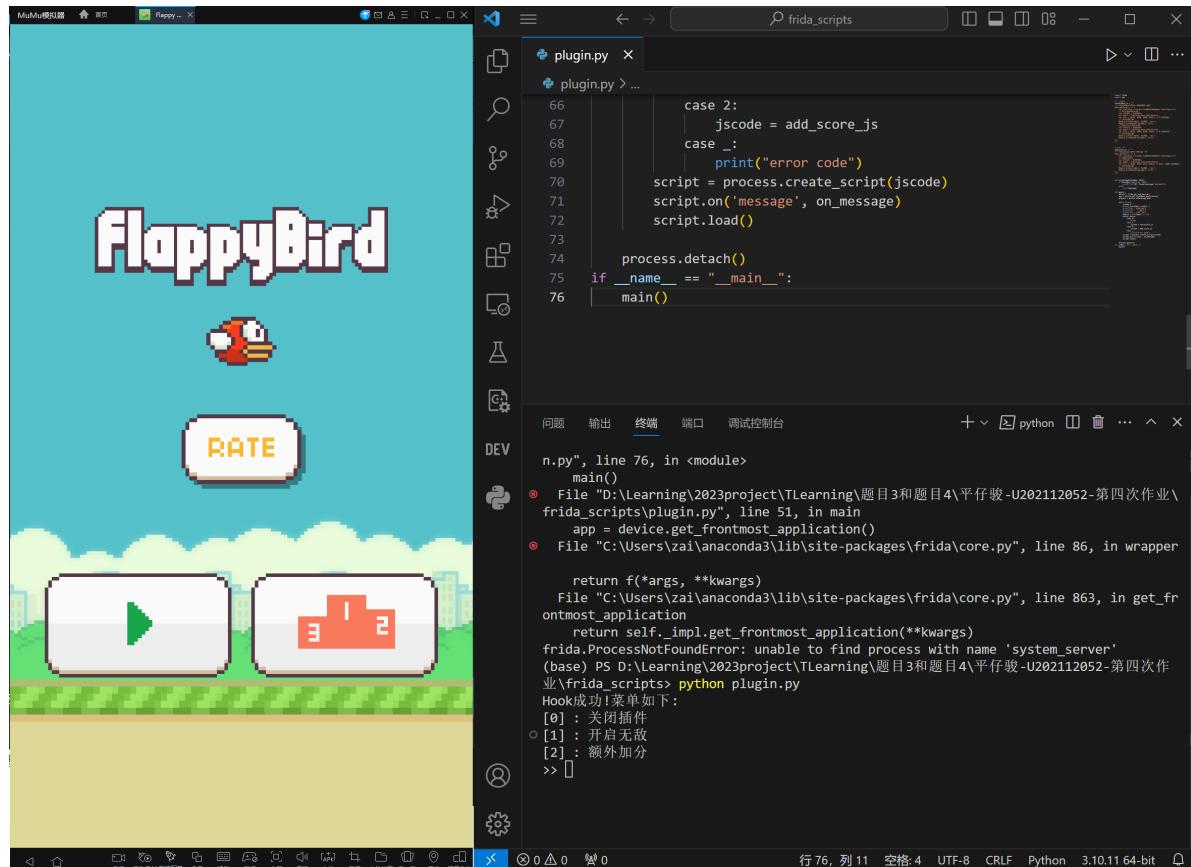
```

```

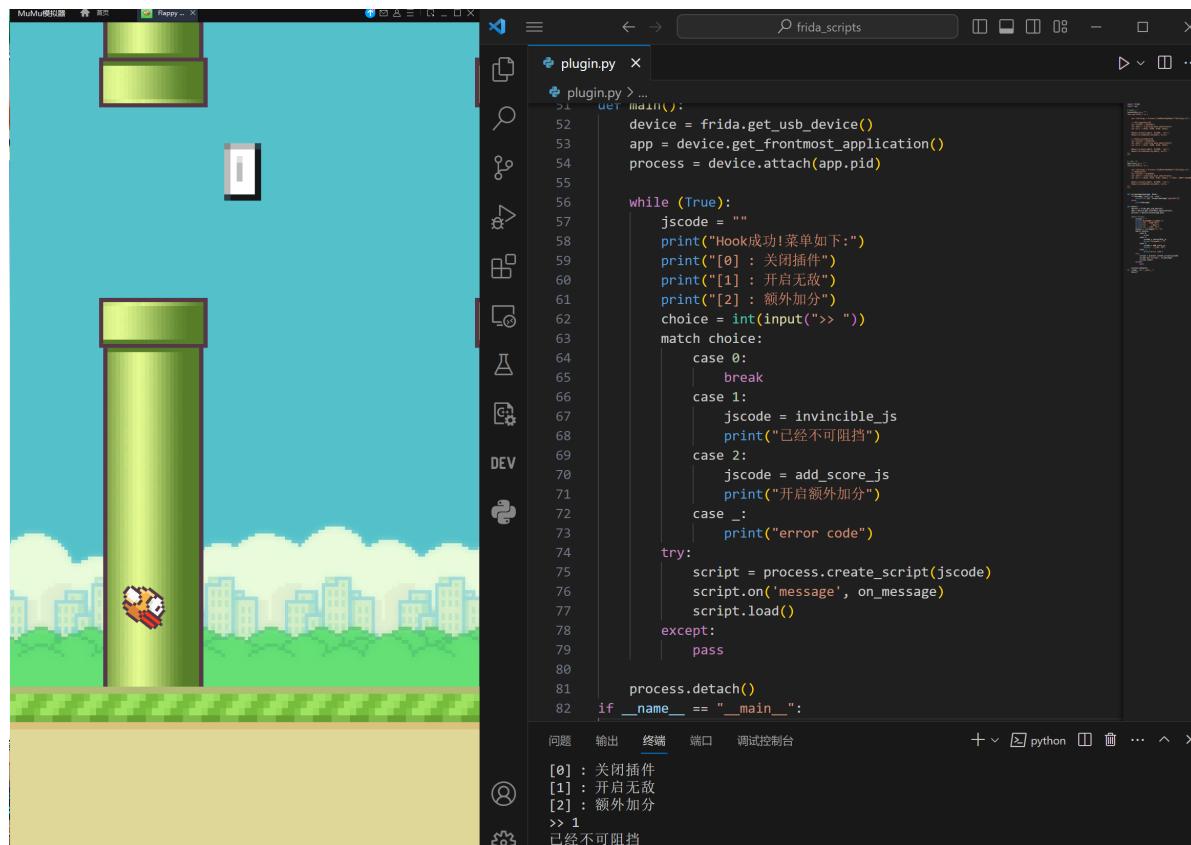
1 // 以下脚本可以实现额外加分
2 Java.perform(() => {
3     var libil2cpp = Process.findModuleByName("libil2cpp.so");
4     // UpdateScore
5     var offset1 = 0x5E0A0C;
6     var addr1 = libil2cpp.base.add(offset1);
7     var arr1 = [0x64, 0x10, 0x81, 0xE2]; // 0x64加100分 0x5E0A0C
8     // 修改内存保护
9     Memory.protect(addr1, 0x1000, 'rwx');
10    Memory.writeByteArray(addr1, arr1);
11 });

```

下面是连接上frida_server后启动脚本的效果



开启invincible无法阻挡



开启add_score额外加分

